# Motion Estimation from Laser Ranging for Autonomous Comet Landing

Andrew E. Johnson and A. Miguel San Martin
Jet Propulsion Laboratory, California Institute of Technology
Mail Stop 125-209, 4800 Oak Grove Drive, Pasadena, CA 91109

## Abstract

*This paper describes an algorithm for translational motion estimation near a comet surface using scanning laser rangefinder data. Our technique is based terrain map generation from rangefinder data followed by terrain map alignment. The output of our algorithm is estimates of rigid translational motion and motion covariance between scans. Our algorithms have been tested using data acquired with a prototype scanning laser rangefinder designed for comet landing, and results indicate motion estimation accuracies of 0.5m over 70m of motion with a processing rate of 4.4 Hz on 10,000 sample range scans.*

## 1 Introduction

Comets are small interplanetary bodies left over from the formation of our solar system. A comet is composed of a solid nucleus and a coma, an expanding cloud of gas sublimating off of the nucleus. Comet nuclei are fragile conglomerates of ice, rocks and complex carbon compounds; this composition results in a very dark and irregular surface. Comet nuclei are typically smaller than 10 km in diameter and are rough on all scales.

Because they contain material that will explain the chemical composition of the early solar system, the space science community is very interested in comets. In particular, returning a sample of a comet nucleus for study on earth would provide valuable insights into early solar system formation. For this reason, NASA and other space agencies are planning multiple missions to comets beginning with near body flybys and culminating with the return of multiple samples from a comet nucleus.

Returning a sample from a comet is a multistage process, and a typical scenario is as follows. First, the spacecraft travels to the comet nucleus and establishes a high orbit around the nucleus (50km). While in orbit, the spacecraft will spend a few months acquire imagery from which scientists on the ground will build a model of the comet nucleus including spin rate, 3-D shape, density, and gravity field. After the model is built candidate landing sites will be selected and closer flybys of these landing sites will be used to improve the model resolution at these locations. A landing site will be selected and the spacecraft will begin its descent to the surface of the comet nucleus.

The irregular shape and the small size of nuclei combined with comet outgassing make the vicinity of comet nuclei very dynamic. Furthermore, the low gravity of the nucleus and the need to acquire a sample from below the surface of the comet necessitates anchoring the spacecraft to the comet surface. Anchoring places limits on the horizontal and vertical velocities of the spacecraft at touchdown. Consequently, accurate estimation of spacecraft attitudinal and positional state relative to the comet surface is required during descent.

To adjust to the dynamic environment and keep the spacecraft on course to the selected landing site, multiple precise target relative maneuvers will be required. The round trip light time to comets prohibits the determination of the necessary trajectory control maneuvers on the ground, so autonomous navigation methods must be used. Standard inertial sensors do not provide the positional accuracy needed for comet landing (e.g., accelerometer errors will grow to the kilometer level over a few hours). One possible solution to the precise positioning problem is to use a scanning laser rangefinder. With minimal processing, rangefinder scans can be used to estimate surface relative motion and detected hazards. Furthermore, rangefinder scans can be matched to a 3-D model of the comet to establish comet absolute position.

This paper describes our algorithm for motion estimation near a comet surface using scanning laser rangefinder data. Our technique is based on first generating terrain maps from rangefinder data followed by terrain map alignment. The output of our algorithm is estimates of rigid translational motion and motion covariance between scans.

Since flight computer processing power is limited, our algorithm has been designed to keep processing to a minimum. During development we investigated multiple techniques for motion estimation from range data. Initially we started with an Iterative Closest Point algorithm (ICP) [1][8], but found that, even with extensive accelerations, this algorithm was too slow given the size of our data sets and the frame rates desired. We considered the algorithm of Horn and Harris [4] but decided that our application did not satisfy the smooth surface and small motion assumptions of their approach. Finally, we decided that since onboard sensors give very accurate estimates of rotational motion (gyros are accurate to 0.01° per second), we can assume that rotational motion is known. Using this assumption, we developed an efficient algorithm for estimating translational motion based on our previous work on image-based feature tracking [5].

## 2 Algorithm

Motion estimation from range imagery takes part in three stages: terrain map generation, terrain map alignment and motion estimation.

### 2.1 Terrain Map Generation

Terrain map generation is the process by which range samples are projected into a grid to form a 2½-D terrain map representation. Scanning laser rangefinders generally have spherical or perspective projection models. Also, scan patterns are not always regular raster scans; spiral and helical scans are common when minimizing scanner power. Nonlinear projection models and irregular scan patterns create an irregular sampling of the surface. If the range samples are used directly, then a time consuming registration algorithm that accounts for the irregular spacing between samples is needed (e.g., ICP). However, by resampling the range samples from each scan to a regular grid in Cartesian space, motion estimation can be posed as an image alignment problem greatly simplifying the underlying algorithms and data structures which will ultimately result in a more efficient algorithm.

A terrain map is a function *Z(r,c)* that encodes elevation on a regular grid. To generate a terrain map, the horizontal size of each grid cell, *s*, and horizontal extent, *H*, of the terrain map must be determined. As shown in Figure 1, these parameters can be determined from the scanner field of view *f*, the average of scan samples across the scene *n*, and the average range to the scene being imaged *R*. In general we set these parameters as follows:

$$H = 2R\tan(f/2)$$
$$s = H/n$$

With these settings, the terrain map will cover roughly the same extent as the scanned data and each grid cell will contain approximately one sample.

Once the terrain map parameters are established, the procedure for terrain map generation is as follows. First, each range sample is converted from scanner angle and range coordinates to Cartesian coordinates *(x,y,z)*. Next, the *(x,y)* coordinates of the sample are used to determine the floating point coordinates *(r,c)* that the sample projects to in the grid cell

$$(r,c) = (y/s + H/2, x/s + H/2)$$

The coordinate relationship between sensor and terrain map coordinates is shown in Figure 1. In general *(r,c)* will fall between discrete grid cells, so, to prevent aliasing, bilinear interpolation is used to update the terrain map. Two arrays are used to perform bilinear interpolation: the elevation accumulator *E(r,c)* and the bilinear weight accumulator *W(r,c)*. For each sample, the four grid cells surrounding *(r,c)* are updated using

$$p = r - \underline{r} \quad q = c - \underline{c}$$
$$E(r,c) += (1-p)(1-q)z \quad W(r,c) += (1-p)(1-q)$$
$$E(r+1,c) += p(1-q)z \quad W(r,c) += p(1-q)$$
$$E(r,c+1) += (1-p)qz \quad W(r,c) += (1-p)q$$
$$E(r+1,c+1) += pqz \quad W(r,c) += pq$$

where $\underline{x}$ is the floor operator. After all samples have been accumulated, the elevation *Z* at each grid cell is determined using

$$Z(r,c) = E(r,c)/W(r,c)$$

Due to the irregular sampling by the scanner, it is possible that a grid cell did not have a sample projected into it and consequently does not have an elevation value. For efficiency during image alignment, it is important that the terrain map be free of holes, especially near the center of the map. A simple interpolation scheme is used to fill any holes. First, hole cells are detected using a modified grassfire transform that detects cells that do not have an elevation but are surrounded by cells with elevation. Next, each hole cell is assigned the average elevation of all neighboring cells that have elevation values. By repeating this process until all hole cells have an elevation value, the holes in the terrain map are filled incrementally. Figure 2 shows a typical range scan, a terrain map before hole filling and a terrain map after hole filling.

To be aligned by our algorithm, two terrain maps must be generated using the same terrain map parameters. Also any rotation between the scans must be eliminated before the scans are aligned. The following procedure is used to generate two terrain maps for alignment. First, the range samples in each scan are converted to Cartesian coordinates. Next, the rotation between the scans (determined from on board gyros) is eliminated by rotating the samples from the second scan into the frame of the first scan. Next, the terrain map parameters are

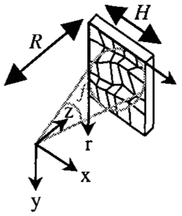
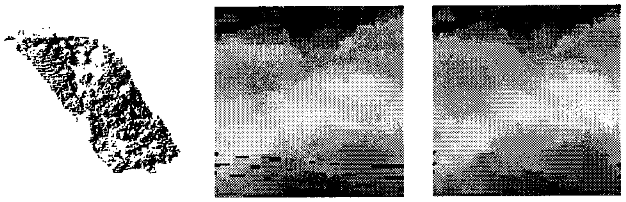


**Figure 1 Sensor and terrain map coordinates.**



3-D range samples　Terrain map with holes　Terrain map with filled holes

**Figure 2 Terrain map generation.**

determined using the data from the first scan. These parameters are then used to generate the terrain maps for both scans ensuring that the size of the grid cells are the same for each image. The end results of terrain map generation are two terrain maps that are ready for terrain map alignment.

## 2.2 Terrain Map Alignment

During terrain map alignment one terrain map is shifted relative to another by $\mathbf{d} = (dr,dc,dz)$ until the difference in elevation data between the two maps is minimized. Our procedure for terrain map alignment is inspired by the Shi-Tomasi feature tracker [7]. However, we modify the tracker to use the additional elevation information to provide full 3-D tracking.

Suppose the terrain map $I(r,c)$ is generated and then, using samples from a later scan, the terrain map $J(r,c)$ is generated. We would like to solve for the 3-D image shift **d** between the scans. Following the derivation of Shi and Tomasi, at the correct shift, the relationship

$$I(r+dr,c+dc)+dz = J(r,c)$$

holds. To constrain the problem so that we can solve for the 3-D shift and account for noise in the data, we seek to minimize

$$\varepsilon = \sum_W (I(r+dr,c+dc)+dz-J(r,c))^2 \qquad (1)$$

over a window $W$ that covers most of the terrain map. The minimum of ε can be found by differentiating ε with respect to the image shift **d** and setting the result to zero

$$\frac{1}{2}\frac{\partial\varepsilon}{\partial d} = \sum_W (I(r+dr,c+dc)+dz-J(r,c))\mathbf{g} = 0 \qquad (2)$$

where

$$\mathbf{g} = (\partial I/\partial r, \partial I/\partial c, 1)\,.$$

Finite differences are used to compute the gradients of the terrain map

$$\frac{\partial I}{\partial r} = \frac{I(r+1,c)-I(r-1,c)}{2} \quad \frac{\partial I}{\partial c} = \frac{I(r,c+1)-I(r,c-1)}{2}$$

If the image shift is small then $I(r+dr,c+dc)+dz$ can be approximated by its truncated Taylor series expansion

$$I(r+dr,c+dc)+dz = I(r,c)+\mathbf{g}^T\mathbf{d} \qquad (3)$$

Substituting (3) into (2) and rearranging terms results in

$$\sum_W (J(r,c)-I(r,c))\mathbf{g} = (\sum_W \mathbf{g}\mathbf{g}^T)\mathbf{d}$$

This is a linear equation in the unknown **d**

$$\mathbf{e} = H\mathbf{d} \qquad (4)$$

$$\mathbf{e} = \sum_W (J(r,c)-I(r,c)\mathbf{g} \quad H = \sum_W \mathbf{g}\mathbf{g}^T$$

Because of the linearization, the solution to (4) does not minimize (1) exactly. However using (4), a Newton Raphson iterative minimization can be used to align the terrain maps exactly. The procedure is to first solve (4) for $\mathbf{d_0}$ (H and **e** are constructed assuming $\mathbf{d} = 0$). Then iteratively solve (4) for $\mathbf{d_i}$ with **e** replaced by

$$\mathbf{e_i} = \sum_W (J(r,c) - I(r+dr_i,c+dc_i) - dz)\mathbf{g}$$

until $\mathbf{d_i}$ changes very little. $\mathbf{d_i}$ is a floating point value, so $I(r+dr_i,c+dc_i)$ is determined through bilinear interpolation of the four neighboring grid cells to $(r+dr_i,c+dc_i)$. The end result of terrain map alignment is a vector **d** that aligns the two terrain maps.

The window $W$ over which the two terrain maps share data, and therefore can be compared, changes at each iteration. It is possible to determine W at each iteration so that all possible data is used. However, if $W$ is fixed for all iterations, a more efficient algorithm results because H is computed only once per alignment.

In order to maximize the overlap between terrain maps and consequently minimize the alignment error, $W$ should be set as large as possible. However, because of boundary effects, it is not possible to set $W$ to the entire terrain map. Ideally, $W$ will be set such that when the terrain maps are aligned using the correct transformation (rotation and translation) $W$ is the largest window contained completely within both maps. Since it is not possible to know the translation between terrain maps before alignment, our algorithm sets $W$ by using a translation extrapolated from the translation computed between the previous two scans. Another alternative is to set $W$ based on a translation predicted from on-board inertial sensors.

## 2.3 Motion Estimation

The purpose of motion estimation is to transform the alignment vector **d** into a 3-D translation **T** and also compute the covariance matrix C of the translation.

**T** can be computed directly from **d** by

$$\mathbf{T} = S\mathbf{d}$$

where

$$S = \begin{bmatrix} 0 & s & 0 \\ s & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Since d is estimated using least squares, the covariance of **d** is the inverse of H. Given that **T** is a linear function **d**, the covariance of **T** can be computed from H as well.

$$C = (S^T HS)^{-1} \sigma^2$$

$\sigma^2$ is the variance on the terrain map noise which can be computed from sensor noise characteristics. Once the translation and translation covariance are computed, they can be passed to the spacecraft guidance, navigation and control subsystem for execution of safe and precise trajectories.

## 2.4 Multi-frame Motion Estimation

During autonomous landing, multiple range scans will be taken as the lander approaches the comet surface. For small translations, alignment errors are roughly independent of the magnitude of translation; in general they are between 1/3 and 1/6 the terrain map grid cell size. If motion estimation is done between subsequent scans, then the motion estimation error will accumulate a fixed error for each scan. However, if the translation between scans is small with respect to the extent of the surface area scanned, then it will be possible to align multiple range scans to a single key scan. In this case, the alignment errors will remain fixed for each key scan resulting in a less rapid growth in alignment errors. At some point, it will become difficult to align a scan with the current key scan because the overlap becomes too small. When this happens, the key scan is updated to the current scan. Although the accumulation of errors cannot be eliminated, this procedure will keep it to a minimum.

Using key scans also has advantages in terms of efficiency. During alignment, terrain map gradients are computed only for the first map. Since the first map corresponds to the key scan, gradients will only have to be computed each time the key frame is changed. Since computing image gradients takes roughly half the total time to estimate motion between two scans, eliminating this step results in an algorithm that is twice as fast.

Deciding when to select a new key scan is not straight forward. This decision depends on the surface overlap between scans and the overall roughness of the surface being scanned. In our algorithm, we select a new key frame based solely on the overlap between scans; when the window of overlap $W$ between scans falls below a threshold based on the number of grid cells in the terrain maps, a new key frame is selected.

Now that we have discussed all of the components, we can describe our motion estimation algorithm in its entirety. The first scan is taken, its terrain map is generated and the gradients of this terrain map are computed. This is the key scan. The next scan is taken and its terrain map is generated using the terrain map parameters of the key scan. The comparison window $W$ is set based on an initial prediction of the translation between scans. Next the terrain maps are aligned and the motion and its covariance between scans are computed. The next scan is read in, and its terrain map is generated using the parameters of the key scan. $W$ is set based on the motion extrapolated from previous alignments. The current terrain map and the key map are aligned. This procedure repeats until $W$ shrinks below ½ the total number of grid cells in the terrain maps; at this point a new key map is selected and the procedure repeats.

# 3 Results

To characterize the performance of the algorithm, scans of outdoor scenes were collected using a long range scanning laser rangefinder.

The sensor used was a scanning laser rangefinder developed for the ST4/Champollion comet lander mission. The rangefinding principal used is pulsed time of flight. The nominal maximum range of the sensor is 500 m, range discretization is 2 cm, and laser beam divergence is 2 mrad. The pulse repetition frequency of the sensor is 9 kHz, so a 100x100 image can be acquired in little over a second of scanning.

Two axis scanning is accomplished using two mirrors to deflect the laser beam. The scanner has a programmable scan pattern. Pointing to a fixed direction is possible as well as raster and spiral scans of variable number of scan lines and field of view. The maximum field of view of the sensor is 10°x10°.

This sensor was taken out into the field to collect realistic comet surface data. Multiple factors contribute to finding a *realistic* comet scene. First of all, comets are believed to be rough on all scales, so scenes with features ranging from hills to rocks to pebbles are desired. Second, the scenes imaged should be free of man-made objects and if possible vegetation. Finally, the data will be collected from the ground, so scenes with a large vertical extent are desirable.

During data collection, a digital theodolite was used to establish ground truth sensor motion. Before data collection the sensor and digital theodolite were rigidly attached to a metal plate. Then, in the laboratory, the rigid transformation between the theodolite and the sensor was computed using a range sensor calibration procedure [2].

At each scanning location, the theodolite and sensor are placed on a tripod, manually leveled, and pointed at surface to be scanned. The theodolite is then used to measure the 3-D position of stationary reflective targets placed in the vicinity of the tripod in order to establish the ground truth position of the sensor. Finally, the desired scans are taken. Once the scans are completed, this process is repeated at the next location along the trajectory of the sensor.

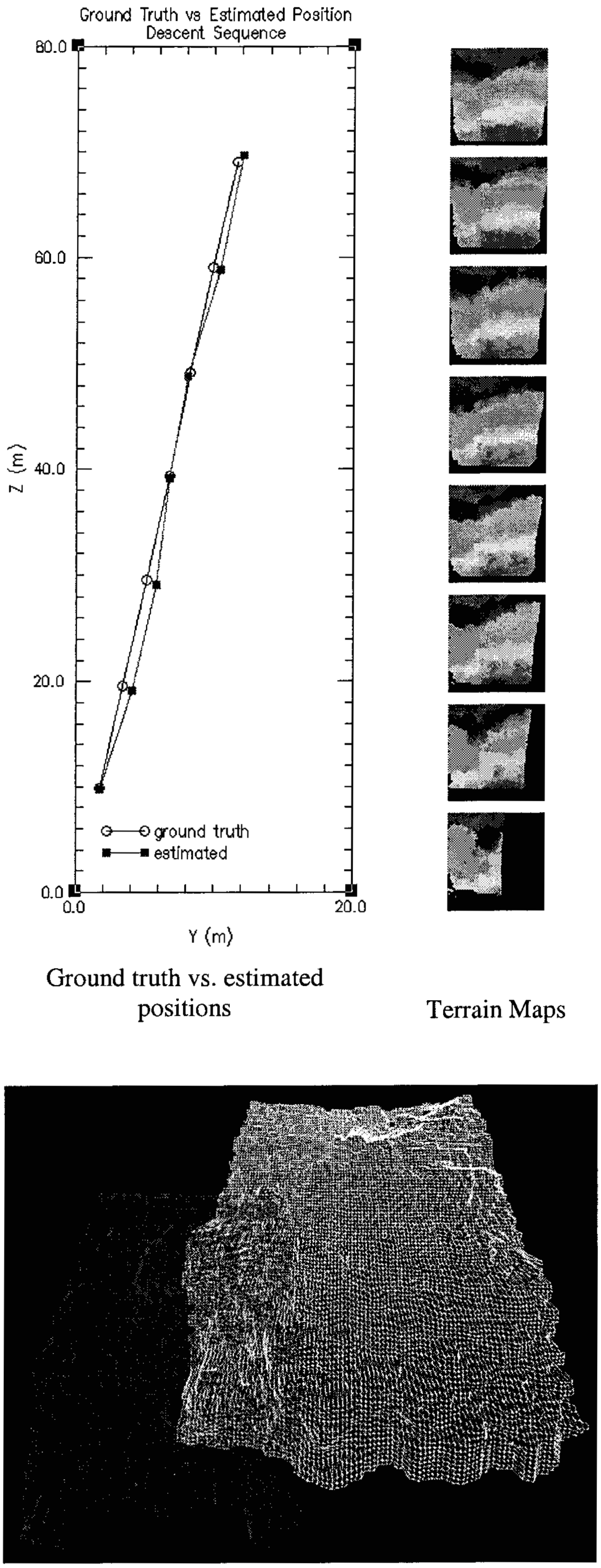
After data collection, the ground truth for the relative 6 degree-of-freedom motion between sensor locations is determined by computing the rigid transformation between corresponding ground truth marker locations using the method of Faugeras and Hebert [3]. This ground truth is used to establish the effectiveness of our translational motion estimation algorithm. It also provides the estimates of sensor rotation used to correct for changes in sensor attitude between scans.

The first data set collected was taken in the JPL East Parking Lot. A steep hill, free of man-made objects, containing rocks and (unavoidably) vegetation was scanned. The trajectory followed a straight path toward the hill, with the pointing target located near the end of the trajectory. The sensor was tilted up approximately 10° with respect to the theodolite to remove the ground from the images. The trajectory started approximately 270 m from the hill and ended approximately 200 m from the hill. Steps of 10 m were taken between each sensor location. Six targets were used to establish ground truth.

At each location a square scan of 100x100 range samples with a scanner field of view of 10° was taken, and from these scans 100x100 terrain maps were generated. The terrain maps generated and the estimated trajectory vs. the measured ground truth are shown in Figure 3. Depth in the terrain maps is color-coded based on the visible spectrum; red is close to the sensor, and magenta is far. Figure 3 also shows the first and last terrain maps in their aligned positions. Table 1 shows the quantitative comparison of ground truth vs. estimated motion. Absolute error is defined as the vector difference between the ground truth translation and the estimated translation. Relative error is absolute error divided by the magnitude of the ground truth translation.

For this data sets the results are promising. Relative errors decrease to less than 1% by the end of the sequence. Furthermore absolute errors are around 0.5 m; for an altitude of approximately 200 m, these errors are on order of 1/2 of the spacing between range samples.

Computing the ground truth motion requires the registration of the target points collected from different sensor positions. Errors in measuring target positions with the theodolite will translate into errors in the estimated motion, so the ground truth computed will not be perfectly accurate. Although there is no absolute way to verify the ground truth, the root mean square error of the registered target points will give an indication of the



Ground truth vs. estimated positions

Terrain Maps

Alignment of first and last terrain maps

**Figure 3 Descent Sequence.**

accuracy of the ground truth. In the descent sequence data set, the RMS error on registered target points was around 0.1 m. This error is significant enough that some of the motion estimation errors can be attributed to errors in estimating ground truth.

The second data set was taken in the at the China Lake Naval Weapons Station. A rocky face free of vegetation and man made objects was scanned. The trajectory followed a straight line 16 m from and parallel to the face with the sensor pointed toward the face. Steps of approximately 0.10 m were taken between each sensor location. Two targets and the leveling of the sensor were used to establish ground truth.

At each location a square scan of 100x100 range samples with a scanner field of view of 10° was taken, and from these scans 100x100 terrain maps were generated. The terrain maps generated and the estimated trajectory vs. the measured ground truth are shown in Figure 4. Figure 4 also shows the first and last terrain maps in their aligned positions. Table 2 shows the quantitative comparison of ground truth vs. estimated motion with the quantities defined in Table 1.

The results for the horizontal sequence are not as good as in the descent sequence. Most likely this is due to the fact that only two ground truth markers were reliably measured causing ground truth determination to be underconstrained. To establish ground truth, an assumption that the sensor was level was used to provide an additional constraint. Because this assumption was only approximately true in reality, the ground truth was not estimated very accurately. Even so, an absolute error of 0.024 m from an distance of 16 m is approximately ½ the spacing between range samples.

Number of range samples and terrain map size are the primary parameters influencing algorithm running times. Both the descent and horizontal sequences are composed of 10,000 range sample scans and 10,000 grid cell terrain maps, so they will have comparable running times. On a Silicon Graphics 174 MHz R10000 O2 workstation, each alignment (including sample rotation, terrain map generation and terrain map alignment) takes 200 ms and each time the key scan is changed, 200 ms are needed to compute gradient maps. The descent sequence has 7 alignments and 1 key scan resulting in an update rate of 4.4 Hz. The horizontal sequence has five alignments and one key scan resulting in an update rate of 4.2 Hz. These update rates are more than adequate for comet landing where scans will be taken at 1 Hz.

## 4 Simulation

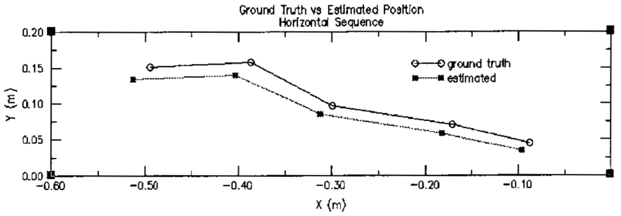
Another approach to evaluating algorithm performance is to use simulated rangefinder data and Monte Carlo

**Table 1 Translation error magnitudes for the descent sequence.**

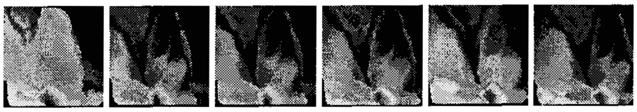
| Location | Absolute Error | Relative Error | Translation Magnitude |
|---|---|---|---|
| 1 | 0.49 m | 4.9% | 10 m |
| 2 | 0.47 m | 2.3% | 20 m |
| 3 | 0.42 m | 1.4% | 30 m |
| 4 | 0.37 m | 0.9% | 40 m |
| 5 | 0.49 m | 1.0% | 50 m |
| 6 | 0.49 m | 0.8% | 60 m |
| 7 | 0.69 m | 1.0% | 70 m |

**Table 2 Translation error magnitudes for the horizontal sequence.**

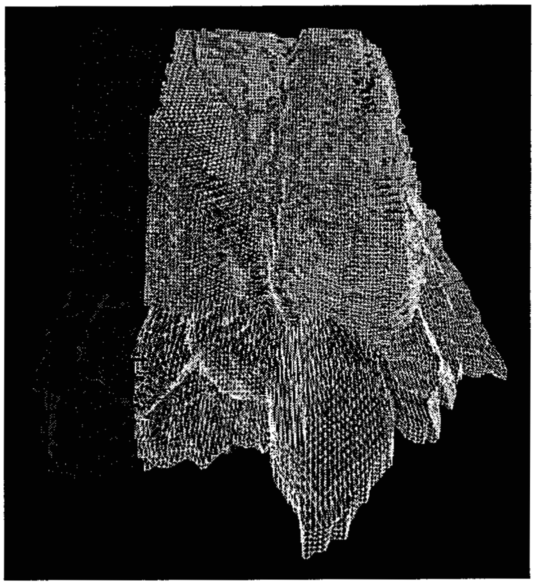
| Location | Absolute Error | Relative Error | Translation Magnitude |
|---|---|---|---|
| 1 | 0.013 m | 14.4% | 0.09 m |
| 2 | 0.017 m | 10.0% | 0.17 m |
| 3 | 0.017 m | 5.7% | 0.30 m |
| 4 | 0.024 m | 6.2% | 0.39 m |
| 5 | 0.024 m | 4.9% | 0.49 m |



Ground truth vs. estimated positions



Terrain maps



Alignment of first and last terrain maps

**Figure 4 Horizontal Sequence**

techniques to establish the expected performance of the algorithm. In addition to establishing expected motion estimation errors, this simulation can be used to design scan acquisition strategies including how often to take scans and when to change key scans.

We have built a laser rangefinder simulation tool. The tool takes as input a synthetic terrain map, the trajectory of the sensor, the sensor scan pattern and other sensor parameter including range accuracy, range discretization, scanner accuracy, scanner discretization, and laser beam divergence. Output are the range samples acquired and a corrupted attitude estimate based on the supplied trajectory. The simulation models the motion of the sensor during scanning. It also simulates the divergence of the laser beam by modeling a laser beam as a bundle of rays that intersect the surface; the returned range is the shortest range of all rays in the bundle.

A motion estimation trial consists of creating a new synthetic terrain map, generating a random trajectory within a class of trajectories (e.g., trajectories of a certain descent angle), synthesizing the scanned range samples and estimating motion. Statistics on the results of multiple trials are then used to characterize the performance of the algorithm for the given set of scanner parameters and class of trajectory.

Using our simulation, we investigated the effect of descent angle on motion estimation errors and how often scans need to be taken. Each trajectory started from an altitude of 100m, with each scan containing 100x100 samples in a 10° field of view, with a range accuracy of 0.02m, a laser beam divergence of 0.1°, and a attitudinal estimation error of 0.01°. The simulation showed that for horizontal motion, motion estimation is accurate to 0.02m over 3.0 m of motion or 0.7%, descent at 45° is accurate to 0.03m over 5m of motion or 0.5%, and straight vertical descent is accurate to 0.04m over 25 m or 0.2%. Our simulation also showed that for pure horizontal motion, images should be taken every 3m, for 45° descent images should be taken every 5m, and for pure vertical descent, images should be taken every 25m. The simulation also estimated the horizontal landing positional accuracy when descending vertically from an altitude of 100m to be 0.16m. If range sensing errors grow linearly with range, the landing error should grow linearly with starting altitude (i.e., 1.6 m landing accuracy can be expected when starting from a known altitude of 1000 m).

## 5 Conclusion

We have developed and tested a motion estimation algorithm that enables autonomous comet landing using scanning laser rangefinder data. We have shown that this form of motion estimation can decrease uncertainty in spacecraft motion to a level that makes landing in the dynamic comet nucleus environment feasible. Our current algorithm assumes that each scan is taken from a fixed position in space. Future work will investigate algorithms for estimating spacecraft motion when the spacecraft is moving during scanning. If this problem can be solved, then it is feasible that scanning laser rangefinders can be applied to the problem of motion estimation during landing planets and outer moons.

## Acknowledgements

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. We would like to thank Curtis Padgett, Daniel Clouse, Aron Wolf and Chester Chu for assisting in data collection, scanner calibration, and providing access to the scanning laser rangefinder.

## References

[1] P. Besl and M. McKay. "A Method for Registration of 3-D Shapes." *PAMI*, 14 (2), pp. 239-256, 1992.

[2] D. Clouse. "Accuracy of the ____ Laser Ranger." JPL Internal Report. October 1999.

[3] O. Faugeras and M. Hebert. "The Representation, recognition and Locating of 3D Shape from Range Data." *IJRR*, 5 (3), pp. 27-52, 1986.

[4] B.K.P. Horn and J. G. Harris. "Rigid Body Motion from Range Image Sequences." *CVGIP: Image Understanding*, 53(1), pp. 1-13, 1991.

[5] A. E. Johnson and L. H. Mathies. "Precise Image-Based Motion Estimation for Autonomous Small Body Landing." *Proc. Int'l Symp. Artificial Intelligence, Robotics and Automation in Space* (iSAIRAS'99), June 1999.

[6] A. E. Johnson and M. Hebert. "Surface Matching for Object Recognition in Complex Three-Dimensional Scenes." *Image and Vision Computing*, vol. 16, pp. 635-651, 1998.

[7] J. Shi and C. Tomasi. "Good Features To Track." *Proc. Computer Vision and Pattern Recognition* (CVPR'94), pp. 593-600, 1994.

[8] Z. Zhang. "Iterative Point Matching for Registration of Free-Form Curves and Surfaces." *IJCV*, 13 (2), pp. 116-152, 1994